Step 1:

pip install unidecode

Step 2: Import dependencies

*# Data Structures*

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **json**

*# Corpus Processing*

**import** **string**

**import** **re**

**import** **csv**

**import** **nltk.corpus**

**from** **unidecode** **import** unidecode

**from** **nltk.tokenize** **import** word\_tokenize

**from** **nltk** **import** SnowballStemmer

**from** **nltk.corpus** **import** wordnet **as** wn

**from** **nltk.stem** **import** WordNetLemmatizer

**from** **sklearn.feature\_extraction.text** **import** TfidfVectorizer

**from** **sklearn.preprocessing** **import** normalize

**from** **collections** **import** Counter

*# K-Means*

**from** **sklearn** **import** cluster

**from** **sklearn.cluster** **import** KMeans

*# Visualization and Analysis*

**import** **matplotlib.pyplot** **as** **plt**

**import** **matplotlib.cm** **as** **cm**

**import** **seaborn** **as** **sns**

**from** **sklearn.datasets** **import** make\_blobs

**from** **sklearn.metrics** **import** silhouette\_samples, silhouette\_score

**from** **wordcloud** **import** WordCloud

*# from jupyterthemes import jtplot*

*# jtplot.style(theme='monokai', context='notebook', ticks=True, grid=False)*

Step 3:

If you don’t have the files in the drive, scrape

pip install Scweet==1.1

**from** **Scweet.scweet** **import** scrape

**from** **Scweet.user** **import** get\_user\_information, get\_users\_following, get\_users\_followers

data = scrap(hashtag="travel", start\_date="2020-04-10", max\_date="2020-04-15", from\_account = **None**,interval=1,

headless=**True**, display\_type="Top", save\_images=**False**, proxy = **None**, save\_dir = 'outputs',

resume=**False**, filter\_replies=**True**, proximity=**True**)

If you have the data files in the drive (Perform this only if you have saved the data files in drive)

**from** **google.colab** **import** drive

drive.mount('/content/drive')

Data = pd.read\_csv('/content/drive/My Drive/tweets/tweets.csv')

Data.head()

Step 4 (optional): Take the histogram to identify the frequency of the lengths

Data['length'] = Data['text'].apply(len)

Data['length'].plot(bins=100, kind='hist')

Step 5: Take the tweets and user IDs

data=Data.copy()

data = data.loc[:,['text',"user"]]

*# data=data.to\_frame()*

data.drop\_duplicates(subset='text', keep=**False**, inplace=**True**)

data.head(6)

tweets\_df=0

data.duplicated(['text']).sum()

Step 6: Take the tweets into a list of text

corpus = data['text'].tolist()

Step 6: Remove numbers and web links, emails and Make all letters lower case

**import** **re**

**def** clean\_text(df, text\_field, new\_text\_field\_name):

df[new\_text\_field\_name] = df[text\_field].str.lower()

df[new\_text\_field\_name] = df[new\_text\_field\_name].apply(**lambda** elem: re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", "", elem))

*# remove numbers*

df[new\_text\_field\_name] = df[new\_text\_field\_name].apply(**lambda** elem: re.sub(r"\d+", "", elem))

**return** df

data\_clean = clean\_text(Data, 'text', 'text\_clean')

data\_clean.head()

Step 7: Download dependencies

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('averaged\_perceptron\_tagger')

nltk.download('wordnet')

Step 8: Tokenized the words and put it in a set

data\_clean['tokenized\_sents'] = data\_clean.apply(**lambda** row: set(nltk.word\_tokenize(row['text\_clean'])), axis=1)

data\_clean.head()

Step 9: Remove all stop words

stopwords = nltk.corpus.stopwords.words("english")

**for** i **in** range(len(stopwords)):

stopwords[i] = stopwords[i].lower()

stopwords[i] = re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", "", stopwords[i])

stopwords[i] =re.sub(r"\d+", "", stopwords[i])

stopwords=set(stopwords)

data\_clean['tokenized\_sents'] = data\_clean.apply(**lambda** row: row['tokenized\_sents']-stopwords, axis=1)

Step 10: Remove all the other words which you has in a text document

other\_words = [line.rstrip('**\n**') **for** line **in** open('/content/drive/My Drive/tweets/stopwords\_scrapmaker.txt')]

**for** i **in** range(len(other\_words)):

other\_words[i] = other\_words[i].lower()

other\_words[i] = re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", "", other\_words[i])

other\_words[i] =re.sub(r"\d+", "", other\_words[i])

other\_words=set(other\_words)

data\_clean['tokenized\_sents'] = data\_clean.apply(**lambda** row: row['tokenized\_sents']-other\_words, axis=1)

data\_clean.head()

Step 11: WordNetLemmatizer has done in this section

wnl = WordNetLemmatizer()

**def** pos\_tag\_text(tokens):

**def** penn\_to\_wn\_tags(pos\_tag):

**if** pos\_tag.startswith('J'):

**return** wn.ADJ

**elif** pos\_tag.startswith('V'):

**return** wn.VERB

**elif** pos\_tag.startswith('N'):

**return** wn.NOUN

**elif** pos\_tag.startswith('R'):

**return** wn.ADV

**else**:

**return** **None**

tagged\_text = nltk.pos\_tag(tokens)

tagged\_text\_set = [(word.lower(), penn\_to\_wn\_tags(pos\_tag))

**for** word, pos\_tag **in**

tagged\_text]

**return** tagged\_text\_set

*# lemmatize text based on POS tags*

**def** lemmatize\_text(tokens):

pos\_tagged\_text = pos\_tag\_text(tokens)

lemmatized\_tokens = [wnl.lemmatize(word, pos\_tag) **if** pos\_tag

**else** word

**for** word, pos\_tag **in** pos\_tagged\_text]

*# print(lemmatized\_tokens)*

**return** lemmatized\_tokens

data\_clean['tokenized\_sents\_1'] = data\_clean.apply(**lambda** row: lemmatize\_text(list(row['tokenized\_sents'])), axis=1)

data\_clean.head()

Step 11: 2 letter and single letter words are moved from the list

alp="qwertyuioplkjhgfdsazxcvbnm"

word2=[i **for** i **in** alp]

**for** i **in** alp:

**for** j **in** alp:

word2+=[i+j]

word2=set(word2)data\_clean['tokenized\_sents\_2'] = data\_clean.apply(**lambda** row: list(set(row['tokenized\_sents\_1'])-word2), axis=1)

data\_clean.head()

Step 12: Join the sets and made a normal text (sentence)

data\_clean["tokenized\_sents\_2"]=data\_clean.apply(**lambda** row: " ".join(row['tokenized\_sents\_1']), axis=1)

data\_clean.head(5)

Step 13:

data=data\_clean.loc[:,["user","id","tokenized\_sents\_2"]]

data.to\_csv("/content/drive/My Drive/tweets/data\_clean.csv")

data.head(5)













